Fvwm: Panels and Tricks (v1.0 2024-10-02)

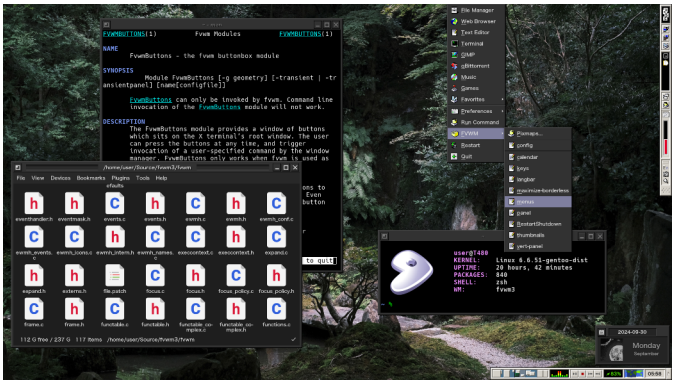
This guide will try to explain how to create a panel in fvwm and some other fvwm tricks.
Be aware that I'm not fully knowledgeable about fvwm, so I'm sure that some of this can be considered bad practices.

You can edit and share this document as you wish. I don't want any credit, but if you upload it or use it for some other resource, then it could be a good idea to say it was made by someone else. The best outcome would be if you create your own panels and modules and share them.

Contents

1. Panels
2. Pixmap borders and colorsets
3. Updating Button Icon
4. Text Shadow and Vertical Spacing
5. Game Configuration

Note: This was made using fvwm3 (1.1.0), but this should mostly work on fvwm2.



Note: Many of the screenshots are from my laptop, so things appear a lot bigger that what they should on a desktop.

Panels

These are some example panels that you can achieve using FvwmButtons and swallowing programs, as described in below. Advanced borders and FvwmScripts won't be covered.

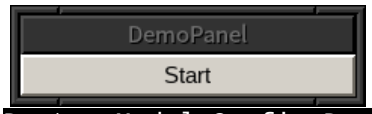
Panel 1: FvwmPager, xosview, stalonetray, cava, media buttons (FvwmButtons), sunclock, clock (FvwmScript).

Panel 2: FvwmPager, cava, media buttons (FvwmButtons), battery (FvwmScript), "fake" sunclock (FvwmButtons), clock (FvwmScript).

Vertical panel: stalonetray, HDD bar (FvwmScript Vdipstick).

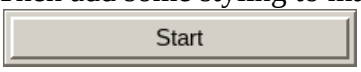
Creating a panel

Let's start by writing a simple panel. I'll add a basic layout.




```
DestroyModuleConfig DemoPanel:*
*DemoPanel: Geometry 222x28-0-0
*DemoPanel: Rows 1
*DemoPanel: Font "xft:Sans:size=9:antialias=True"
*DemoPanel: Colorset 50


# Panel items
*DemoPanel: (Title "Start")
```

Then add some styling to make it look and function like a panel.




```
# Panel style
AddToDecor PanelDecor
+ BorderStyle -- HiddenHandles NoInset

#Style "DemoPanel" FixedSize, FixedPosition
Style "DemoPanel" UseDecor PanelDecor, BorderColorset 14, HilightBorderColorset 14,
NoTitle, Sticky, WindowListSkip, NeverFocus, BorderWidth 5, HandleWidth 5,
FvwmBorder
```

The "FixedSize, FixedPosition" style is nice if you don't want the panel to be scaled or moved.

Next, I'll add some more buttons.

```
# Panel items
*DemoPanel: (20x1, Title "Start", Frame 2)
*DemoPanel: (1x1, Frame 0)
*DemoPanel: (20x1, Title "Status", Frame -2, Colorset 51)
*DemoPanel: (1x1, Frame 0)
*DemoPanel: (30x1, Swallow MediaKeys "Module FvwmButtons MediaKeys", Frame -2)


*MediaKeys: Colorset 50
*MediaKeys: Frame 2
*MediaKeys: Rows 1
*MediaKeys: Padding 0 0
*MediaKeys:  (Action(Mouse1) "Exec playerctl play-pause", Icon "pixmaps/media-play-
pause.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl stop", Icon "pixmaps/media-stop.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl previous", Icon "pixmaps/media-
prev.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl next", Icon "pixmaps/media-next.png")
```

Note: I placed the images in ".fvwm/pixmaps", but it could be a good idea to place them together with the panel configuration in something like ".fvwm/modules/demo-panel/...".

In this example I set "Rows" to 1 and left "Columns" unset. The reverse option can be used for a vertical panel. Each of the "20x1" lines defines each button width in no relation to the panel size. Fvwm resizes them relative to each other. See "*FvwmButtons: BoxSize algorithm" in the "fvwmbuttons" manual for changing this behavior.

As you can see, each "*DemoPanel: ..." line adds a new item. The (1x1, Frame 0) entries act like spacers.

"MediaKeys" is a panel just like DemoPanel and also can be added to the start function. But in this example it's swallowed by the panel. "Padding 0 0" makes the icons stay centered and without clipping. You can also add lines like this to each button individually.

"Frame x" appears a lot. This adjusts the bevel size. You can also swallow a lot of panels inside each other with different frame sizes to get some interesting results.

Other programs can be swallowed with:

```
*DemoPanel: (10x1, Colorset 16, Swallow (NoClose, UseOld) "xclock" "Exec exec
xclock", Frame -2)
*DemoPanel: (32x1, Colorset 20, Swallow FvwmScript-DateTime 'Module FvwmScript
FvwmScript-DateTime', Frame -2, Action(Mouse 1) "ToggleCalendar")
```

Next I'll add a background color to the status panel.

I have some trouble setting the background color on individual buttons. This solution abuses colorsets and an is not ideal. The "fvwmbuttons" man page has more info about this. Maybe "Back ..." or "Container" can be used.

```
# Colorset
InfoStoreAdd color-demopanel "#d4d0c8"
InfoStoreAdd color-demopanel-hi "#ffffff"
InfoStoreAdd color-demopanel-sh "#6d6961"

# Panel
Colorset 50 fg #000000, \
      bg $[infostore.color-demopanel], \
      hi $[infostore.color-demopanel-hi], \
      sh $[infostore.color-demopanel-sh]

# Panel status (normal borders, black background)
Colorset 51 fg #fff, \
      bg #000, \
      hi $[infostore.color-demopanel-hi],
      sh $[infostore.color-demopanel-sh]
```

You should of course use your own colorset values. 50 and 51 is just what I picked for this example.

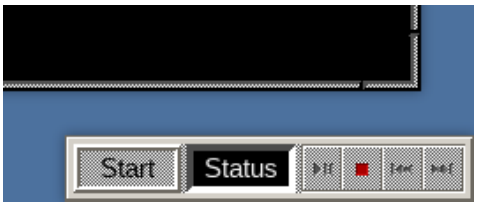
Full configuration:

```
DestroyModuleConfig DemoPanel:*
*DemoPanel: Geometry 222x28-0-0
*DemoPanel: Rows 1
*DemoPanel: Font "xft:Sans:size=9:antialias=True"
*DemoPanel: Colorset 50

# Panel style
AddToDecor PanelDecor
+ BorderStyle -- HiddenHandles NoInset

#Style "DemoPanel" FixedSize, FixedPosition
Style "DemoPanel" UseDecor PanelDecor, BorderColorset 14, HilightBorderColorset 14,
NoTitle, Sticky, WindowListSkip, NeverFocus, BorderWidth 5, HandleWidth 5,
FvwmBorder

# Panel items
*DemoPanel: (20x1, Title "Start", Frame 2)
*DemoPanel: (1x1, Frame 0)
*DemoPanel: (20x1, Title "Status", Frame -2, Colorset 51)
*DemoPanel: (1x1, Frame 0)
*DemoPanel: (30x1, Swallow MediaKeys "Module FvwmButtons MediaKeys", Frame -2)

*MediaKeys: Colorset 50
*MediaKeys: Frame 2
*MediaKeys: Rows 1
*MediaKeys: Padding 0 0
*MediaKeys:  (Action(Mouse1) "Exec playerctl play-pause", Icon "pixmaps/media-play-
pause.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl stop", Icon "pixmaps/media-stop.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl previous", Icon "pixmaps/media-
prev.png")
*MediaKeys:  (Action(Mouse1) "Exec playerctl next", Icon "pixmaps/media-next.png")


# Colorset
InfoStoreAdd color-demopanel "#d4d0c8"
InfoStoreAdd color-demopanel-hi "#ffffff"
InfoStoreAdd color-demopanel-sh "#6d6961"

# Panel
Colorset 50 fg #000000, \
      bg $[infostore.color-demopanel], \
      hi $[infostore.color-demopanel-hi], \
      sh $[infostore.color-demopanel-sh]

# Panel status (normal borders, black background)
Colorset 51 fg #fff, \
      bg #000, \
      hi $[infostore.color-demopanel-hi],
      sh $[infostore.color-demopanel-sh]
```

Again, experiment with frames, swallowed panels, colorsets and icons.

Pixmap Borders and Colorsets

```
# Colorset with tiled pixmap:
Colorset 50 fg #000, bg d4d0c8, TiledPixmap "dither.png"

# Border with tiled pixmap:
BorderStyle TiledPixmap "dither.png"
```



This example doesn't look nice, but this can be useful for certain styles.

Updating Button Icon

You can update a button icon if you add an "Id" tag to it.

```
*Langbar:  (1x14, Id "A", Icon "pixmaps/langbar/en.png", ...)

# Call to change...
SendToModule Langbar ChangeButton 'A' Icon ...
```

Remember that you can execute fvwm commands from a shell script with "FvwmCommand".

Text Shadow and Vertical Spacing

From: man fvwm3, “Font Shadow Effects”

Text Shadow

The following can be done to add shadow to text.

```
Style * Font "Shadow=1 SE:xft:Sans:size=9:antialias=True"
```

The shadow color is configured with the “fgsh” option in a colorset:

```
Colorset 1  fg #777, bg #323232, fgsh #000
```

Vertical Spacing

Note: This is a hacky solution and should be avoided if possible, by using good fonts with a suitable "TitleStyle Height".

Shadows can also be used to vertically align text. The shadow color (fgsh) must be the same as the text background.

```
MenuStyle * Font "Shadow=4 N:xft:.Hiragino Kaku Gothic Interface:style=W4:size=9"
Colorset 5  fg #777, bg #323232, fgsh #323232
```

Game Config

As a bonus, here are some settings that can improve fullscreen gaming.

```
# Remove titlebars from windows that request it.
Style * MWMDecor

# Respect "mwm prohibited operations".
Style * MWMFunctions
Style *.exe !MWMFunctions

# Windows like window placement.
Style * CascadePlacement

# Click to focus and raise windows.
Style * ClickToFocus, MouseFocusClickRaises

# Focus new windows.
Style * FPGrabFocus

# This can fix wine window layering issues.
Style * FPLenient

# Dialog windows
Style * StackTransientParent, DecorateTransient
Style * FPGrabFocusTransient, FPReleaseFocusTransient
#Style * DontStackTransientParent

# If the taskbar is above the game, then EWMHUseStackingOrderHints can reportedly
fix it.
#Style * EWMHUseStackingOrderHints
```

If that isn't enough, then the following forces borderless "*.exe" programs to maximize. Be aware that this also maximizes game launchers and other borderless windows. It can also cause some unpredictable fullscreen behavior (try it yourself).

```
# Maximize: True (not toggle) fullscreen (only if the window supports fullscreen)
DestroyFunc ExeFullscreenMaximize
AddToFunc   ExeFullscreenMaximize
+ I Maximize True fullscreen ewmhiwa
+ I Move 0 0

# Test and maximize window.
DestroyFunc ExeFullscreenTest
AddToFunc   ExeFullscreenTest
+ I ThisWindow ("*.exe", !Borders, !HasTitle) ExeFullscreenMaximize

# Event on "configure_window".
DestroyModuleConfig ExePlaceEvent: *
*ExePlaceEvent: configure_window ExeFullscreenTest

KillModule FvwmEvent ExePlaceEvent
Module FvwmEvent ExePlaceEvent
```